REMARKS

The above amendment and these remarks are responsive to the Office Action of Examiner James H. Blackwell, dated 10 Aug 2005, and designated FINAL.

Claims 1, 5-12, and 14-19 are in the case, none as yet allowed.

35 U.S.C. 103

Claims 1-6, 11, 13-19 have been rejected under 35 U.S.C. 103(a) over Cannata et al. (U.S. Patent No. 6,917,962, hereinafter, Cannata) in view of Salas et al. (hereinafter Salas, U.S. Patent No. 6,233,600).

Applicants have canceled claims 2, 3, 4, and 13.

Applicants invention relates to the creation of forms for the purpose of collecting data from users of collaboration space to create content to add to the collaboration space, and not for the creation of the space itself.

With respect to claims 1-6, 11, and 13-19, the Examiner refers to Cannata [Col. 3:46-50] for teaching "preparing... a form containing fields". As is apparent from the Abstract, Cannata, teaches the use of a standard form ("site builder") to create a work group. That is, Cannata relates to use of a standard form to create a site, but does not teach creation of forms by a user for entry of data content to such a site, including entry of such data by other users of the site.

Applicants have amended the independent claims 1, 11, and 13-19 to clarify this distinction with respect to Cannata.

Similarly, with respect to claims 1-6, 11, and 13-19, the Examiner cites Salas as teaching allowing "the user to modify templates that contain instructions leading to the rendering of HTML pages once the pages are uploaded to the server and processed. [Col. 16:57-64]. While Salas teaches providing a networked collaborative work environment, referred to as an eRoom, and discusses the use of HTML, Salas does not provide for the user to create (as distinguished from use) HTML forms, as do applicants, to define the data model of collaboration space by automating the creation of fields within collaboration space for use by users of collaboration space for creating data content. While Salas uses templates to render content, processing

data so that it can be displayed, he does not teach creating forms for content collection from end users. He does allow users to create (as distinguished from use) templates for uploading content in the form of files, but does not render to the user fields (blank spaces) to be filed in by the user to create content to be added to collaboration space. Thus, Salas teaches storing files, but not the use of user defined forms for collecting content from users.

Further, with respect to claims 1-6, 11, and 13-19, the Examiner refers to Cannata [Col. 5, lines 13-19] for teaching "parsing said form to identify said fields and incorporate them into said schema". Applicants traverse on this point.

Applicants agree that Cannata teaches the creation of a dedicated site. However, in Cannata the same form is used repetitively in creating such sites, and there is no need in such a situation, and therefore no teaching in Cannata that such a form must be parsed. In applicants' invention, because various forms may be created and loaded to collaboration space, the form received must be parsed so that data fields, or data schema, can be deduced and incorporated into that collaboration space for use thereafter in receiving newly defined data content.

This is how applicants' invention works:

1. The user creates (not merely uses) an HTML form, which he drops into an upload control at his browser.

2. The server receives and parses the HTML form to determine which data fields need to be added to collaboration space.

3. The HTML is retained by the server for rendering the form in edit mode to the user for entry of data content to collaboration space.

Thus, an aspect of applicants' invention is the use of the HTML form: the HTML form is analyzed (that is, parsed) to deduce a data model that is used to augment the capabilities of the collaboration space. For example, if parsing the form identifies a name field and an address field as data elements in the original HTML form, the server determines to thereafter monitor name and address fields in the collaboration space.

These features of applicants' invention may be understood by reference to the following material from their specification.

In accordance with a further embodiment of the invention, a review form may be designed in HTML separate from QuickPlace. The resulting form is then

dragged and dropped into QuickPlace, which creates a form for it. This is done by creating a field for each html tag. Thus, each HTML field is parsed to create a corresponding QuickPlace field.

For HTML files, the HTML file is parsed, the linked images found, and the URLs processed. The original file, linked files, and the resulting HTML are then saved on the page with the HTML displayed in read mode, and the original file in edit mode. [Application, page 66, line 11 to page 67, line 2, emphasis added.]

In accordance with a preferred embodiment of the invention, users are provided with a method for defining forms to create pages within collaboration space. These methods include options to upload a document and send a notification, add a meeting to the calendar, or add a task into the QuickPlace.

By clicking on New..., the user gets a list of forms included in QuickPlace that can be used to add a new document to it. The forms provided are sufficient for many uses, but do not give any task-specific ways of adding content to the QuickPlace. To do this, a user may create her own form and adapt it to her particular needs.

There are three ways to create forms: create a form using standard QuickPlace fields; import a form 250 created in Microsoft Office 228; and import a form 122 created in an HTML editor 124. [Specification, page 135, lines 4-19, emphasis added.]

... For example, in a default QuickPlace, a user can create a new QuickPlace form 178. The user chooses which fields to include in form 178, in what order they should appear and what text and or graphics should appear near them. To create this sort of instant structure on the Web using Domino Forms would be very complex indeed. QuickPlace has extended this concept of being able to use HTML to define forms 178 by enabling the creation of custom QuickPlace forms using imported HTML 122. These Forms not only make use of Web authoring technologies such as JavaScript, but also have the back end support of Domino. This back end logic is implemented via tools such as PlaceBots (Domino Agents) 184. This means that forms 178 have the ability to not only to define the look and feel of visible parts of an application, they also have the potential to initiate workflow and many other powerful automated features.

QuickPlace forms 178 have been optimized by stripping away many of the Notes features not required

when used on the Web. A another advantage of this structure is that it enables the use of Web authoring tools to extend the objects. For example, with respect to QuickPlace forms, it is possible to modify forms using XML, JavaScript and HTML and any other Web tools. [Specification, page 22, lines 1-24. Emphasis added.]

A form object 178 is a document used to create new QuickPlace content. The Domino equivalent is a data note of type "h_Form". Form object 178 is a resource used to create, manage and display content, therefore defining the schema of the application. Forms contain fields to hold data, therefore creating and displaying content. Forms can also contain scripts within them to provide logic within the Page. For example, a form can contain form validation to make sure that a field contains only numbers. Forms can also initiate processes outside the page. This is done by creating a PlaceBot 184 and associating the PlaceBot with a Form 178. PlaceBots 184 are not contained by the Form but there is a association between them. [Page 36, line 29 to page 37, line 7. Emphasis added.]

Field object 180 is used to construct (HTML formatted) input fields in forms 178. The Domino equivalent is a Data note of type "h_Field". Fields are constructed from the Domino Form "h_PageUI" with a

the field h_Type set to "h_Field".

Applicants invention create a file which is an HTML file. The server interprets this syntax to create in collaboration space a representation of the semantic of the original form. Applicants don't just take an HTML form and display it. What is different from Salas and Cannata is that the form is used to collect the data and have this data made an element of the collaboration space.

With respect to claims 2, 3, and 4, applicants have canceled the claims.

With respect to claim 5, Salas is cited for teaching a collaborative work environment (collaboration space) including a hierarchical, object based (object model) file structure including pages, folders, and files. Applicants assert that neither Salas nor Cannata teach that collaboration space can have an object model. Wile they do have files and folders, applicants claim that their collaboration space is based upon an object model with a specific semantic, rather than generic objects as suggested by the Examiner. In applicants' collaboration space, the specific object model set forth in claim 5 is used throughout the programming model. Applicants have amended the claim to recite that this specific model "consists" of the objects listed in the claim.

With respect to claim 6, this claim depends from claim 1, and is distinguished from Cannata and Salas as previously discussed with respect to the parent claim.

With respect to claims 11 and 17-18, applicants have amended the claims as previously described. Further, the Examiner appears to have missed an additional limitation. That is, when applicants system and method processes forms that are uploaded by the user, two separate form templates are generated: one to display in read mode and one to display in edit mode, allowing the user to edit the content of that form.

Applicants have canceled claim 13.

Applicants urge that claims 1, 5-6, 14-16, and 19 be allowed.

Claims 7-9 have been rejected under 35 U.S.C. 103 over Cannata in view of Salas, and further in view of Hanson et al. (hereinafter Hanson, U.S. Patent No. 5,956,736).

Salas and Cannata have been discussed with respect to the base claims from which claims 7-9 depend.

Hanson describes an object-oriented editor for creating world wide web documents. Cannata, Hanson and Salas do not

provide for defining the schema of collaboration space by parsing an uploaded html file to identify fields defining that schema and to create therefrom a form for entry in edit mode of data content to the collaboration space.

Applicants urge that claims 7-9 be allowed.

Claims 10-12, and 17-18 have been rejected under 35 U.S.C. 103(a) over Cannata in view of Salas and further in view of Kagle (U.S. Patent No. 6,779, 153).

How claims 11, 17, and 18 distinguishes Salas and Cannata has been previously discussed. Claim 10 depends from claim 1, and claim 12 from claim 11, and each distinguish Salas and Cannata as described previously with respect to their respective base claims.

The Examiner does not apply Kagle to claims 17 and 18.

Kagle has been cited with respect to claim 10 and 12, but not to claims 17 and 18, for its teaching that pictorial/image information can be entered as a pointer to a locally stored image, and for the creation of template files on a client computer. Applicants amended claims distinguish Kagle and Salas inasmuch as neither provides for defining the schema of collaboration space by parsing an uploaded html file to identify fields defining that schema.

With respect to claim 12, which depends from claim 11, resources referenced by a form are obtained and uploaded to collaboration space where they are made available for other uses. Doing so requires that the form be parsed to identify such resources and link the form to instantiations of those resources which now reside at collaboration space, as distinguished from the file system where the form was created.

The Examiner refers to Kable. Kagle's template mapping file 714 is not the equivalent of applicants' form. That is, while Kagle appears to teach the concept of uploading resources that are needed, it does not teach the adjustments necessary to an existing form to reflect the new location of the resources referred to by the form and which were uploaded with the form into collaboration space. Applicants have amended claim 12 to reflect this distinction.

Applicants urge that claims 10-12 and 17-18 be allowed.

SUMMARY AND CONCLUSION

Applicants urge that the above amendments be entered and the case passed to issue with claims 1, 5-12, and 14-19 be allowed.

If, in the opinion of the Examiner, a telephone conversation with applicant(s) attorney could possibly facilitate prosecution of the case, he may be reached at the number noted below.

Sincerely,

Julio Estrada, et al.

By

Shelley M Beckstrand
Reg. No. 24,886

Date: 26 Sep 2005

Shelley M Beckstrand, P.C.
Patent Attorney
61 Glenmont Road
Woodlawn, VA 24381

Phone: 276-238-1972